Real-Time Collision Avoidance with Whole Body Motion Control for Humanoid Robots

Hisashi Sugiura, Michael Gienger, Herbert Janssen and Christian Goerick

Abstract— **We propose a self collision avoidance system that superposes trajectories in order not only to protect the robot's hardware but also to enable continuous motions. The system runs in real-time so that the robot can work in an uncertain environment. It is based on virtual forces between close segments of the robot. The avoidance movements are blended with a whole body motion control in order to change the priority between target reaching and collision avoidance. The blending is performed autonomously without the necessity of external switching. Our method works both while the robot is standing and walking. Reaching motions from the front to the side of the body without the arm colliding with the body are possible. Even if the target is inside the body, the arm stops at the closest point to the target outside of the body. Our method can be used for other applications: We apply it to realizing a "body schema" and for "occlusion avoidance."**

I. INTRODUCTION

Traditionally, targets for robots and the corresponding trajectories were given by operators. They sometimes perform an emergency stop by freezing all motions in the case two segments of the robots come too close. However, nowadays the robots, in particular humanoid robots, are expected to work outside of fixed environments such as factories. They have to interact with dynamic environments where the robots' motions are unpredictable. Thus, we need a more advanced safety mechanism - collision avoidance - instead of an emergency stop. The advantage of such a collision avoidance is not only safety, but it also keeps the robot continuously in motion and can expand its working range.

Collision avoidance methods can be roughly divided into two categories: The first one comprises planning methods that generate trajectories taking known predictable obstacles into account. Many researchers use planning methods and apply them to mobile robots or intelligent industrial robots [1], [2]. Kuffner et al. have proposed a collision avoidance for a humanoid robot [3]. The method comprises a fast collision detection and a real-time planning for gait generation taking into account the possibility of leg interference. They have also proposed a dynamics-based, collision-free planning method using "Rapidly Exploring Random Trees"(RRTs) [4].

On one hand, these methods perform well to generate an globally optimal trajectory in static environments. On the other hand, they are hard to apply to non-predetermined movements due to the computational costs. Because the global trajectory must be re-generated each time the target or the environment of the robot changes. In particular, for robots that have a lot of degrees of freedom (DOFs) such as humanoid robots the computational costs are high.

The other category comprises real-time (reactive) collision avoidance. Methods from this category most commonly superpose simple trajectories such as line segments connecting the current and the target position. The superposed trajectories are not always optimal but they are capable of quickly avoiding obstacles in uncertain environments because they work reactively. For these methods, it is necessary to decide how to avoid and how to change the priority between target reaching and collision avoidance motions depending on the level of risk of collisions in real-time. For this purpose task-prioritizing methods have been proposed that use nullspace optimization criteria [5]–[7]. Brock et al. have proposed collision avoidance for a one arm mobile robot using virtual rubber bands [8].

Seto et al. have proposed a collision avoidance system on a robot with wheels and two arms for the interaction between humans and the robot [9], [10]. The robot segments are modeled with elastic elements that generate virtual forces. This method has been tested on the robot between a static upper body and moving arms.

We have also proposed a collision avoidance method using nullspace optimization criteria and task intervals [11]. However, it is based on 3-D position control for the collision avoidance resulting in a insufficient balance between target reaching and collision avoidance.

The method we propose in this paper is to blend the joint velocity vector of collision avoidance and target reaching motions depending on the distance between the closest segment pair. The collision avoidance uses only 1 DOF and other DOFs are available to be used for target reaching motions.

As far as the authors know, this is the first real-time collision avoidance on a humanoid robot.

II. DISTANCE COMPUTATION

For collision avoidance, it is necessary to compute distances and closest points between segments, i.e. the physical links separated by joints. Many methods for this have been proposed not only in robotics but also in computer games based on accurate models using, for instance, convex hull triangle methods [3], [12]–[15].

However, it is computationally expensive to compute distances and closest points for all possible segment pairs of humanoid robots with accurate models especially on on-board computers. We therefore define the collision model

The authors are with the Honda Research Institute Europe GmbH, Carl-Legien Strasse 30, D-63073 Offenbach/Main, Germany hisashi.sugiura@honda-ri.de

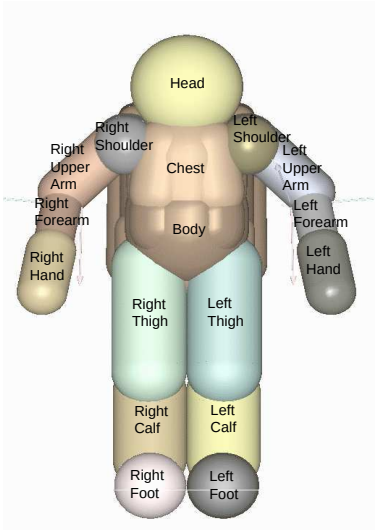


Fig. 1. The collision model is composed of 17 segments. Each segment is composed of one or several sphere-swept-lines or spheres. Different colors were used for visibility.

with primitive objects that can be computed faster as in Fig. 1. Each segment (*Head*, *Right Shoulder* and so on) is composed of spheres or swept sphere lines in order to cover the shape of the robot [16]. Most of the segments are composed of one primitive object but the body and the chest use multiple primitive objects. We compute distances and closest points based on this model.

III. Collision Avoidance

A. Overall concept

The fundamental concept of our method is that motions are continuously blended between the collision avoidance and the whole body motion towards the target. The distributed ratio for two motions is determined with a *danger measure* depending on the distance between closest segment pair and it changes the priority for resulting motion between two motions. The resulting motion is described as,

$$\dot{\boldsymbol{q}} = f(d)\dot{\boldsymbol{q}}_{ca} + \{1 - f(d)\}\,\dot{\boldsymbol{q}}_{wbm} \tag{1}$$

where $\dot{\boldsymbol{q}}$ is the resulting joint velocity vector of the real robot, $\dot{\boldsymbol{q}}_{ca}$ is the joint velocity vector of the collision avoidance control, $\dot{\boldsymbol{q}}_{wbm}$ is the joint velocity vector of the whole body motion control towards the target, $f(d)$ is the blending coefficient and d is the distance between closest segments.

The system architecture is depicted in Fig. 2. The outputs of the collision avoidance control and the whole body motion control are blended with a function depending on the closest distance. Each variable is explained in detail in the next sections.

B. Collision avoidance control

The collision avoidance control uses nullspace optimization criteria to control 2 motions: collision avoidance motion in task space and target reaching motion in nullspace [17]. The task space is used for collision avoidance motions and

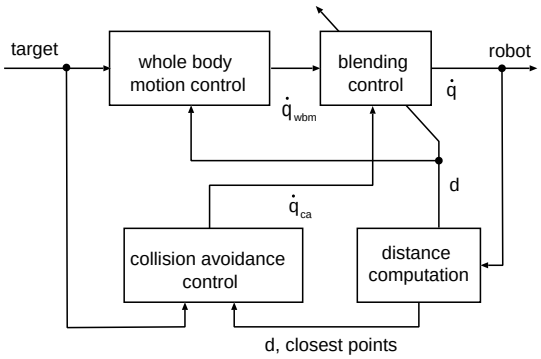


Fig. 2. Our system of the whole body motion control and the collision avoidance

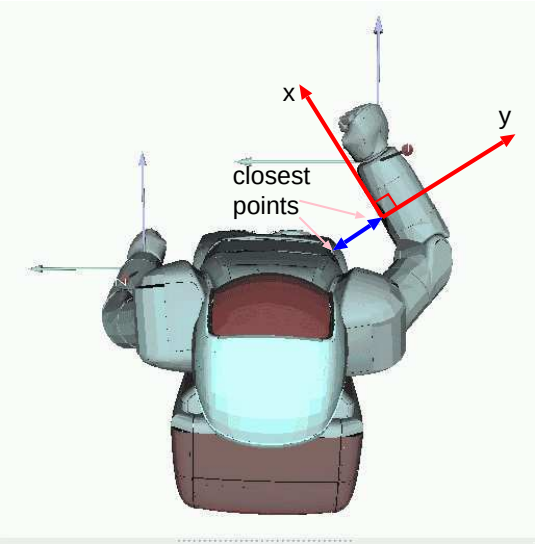


Fig. 3. The collision avoidance coordinate system. The origin is the closest point on the lower arm. Axis y is aligned to a line that connects closest points. Axis z is the outer product of y and a vector that is from the elbow to the wrist position. Axis x direction is outer product of z and y axis.

nullspace is used for target reaching motions, employing a potential function. In this control method, the joint velocity vector $\dot{\boldsymbol{q}}_{ca}$ is computed as

$$\dot{\boldsymbol{q}}_{ca} = \boldsymbol{J}_{ca}^{\#}(\boldsymbol{q})|_{row,y} F_{virtual} + \boldsymbol{N}_{ca}\boldsymbol{\xi}_t, \tag{2}$$

where $\boldsymbol{J}_{ca}(\boldsymbol{q})$ is the collision avoidance Jacobian between closest points and $\boldsymbol{J}_{ca}^{\#}(\boldsymbol{q})|_{row,y}$ is the row vector extracted from the pseudo inverse of Jacobian $\boldsymbol{J}_{ca}(\boldsymbol{q})$. This is the coefficient of the y axis in the collision avoidance coordinate system in Fig. 3. For the collision avoidance purpose only, the y direction needs to be controlled, while the x and z directions of the target motion are left undisturbed. Thus the collision avoidance effectively affects only 1 DOF.

A virtual force $F_{virtual}$ is generated between closest points on closest segments depending on their distance d. Let k be the spring coefficient.

$$F_{virtual} = \begin{cases} k(d_a - d) & \text{if} \quad d < d_a, \\ 0 & \text{otherwise,} \end{cases} \tag{3}$$

where the distance d_a determines a so-called *yellow zone* in which the joint velocities are blended according to Eq. (1).

The $\boldsymbol{N}_{ca}$ is a matrix that maps the vector $\boldsymbol{\xi}_t$ into the nullspace of the task,

$$\boldsymbol{N}_{ca} = \boldsymbol{I} - \boldsymbol{J}_{ca}(\boldsymbol{q})^{\#}\boldsymbol{J}_{ca}(\boldsymbol{q}), \tag{4}$$

where $\boldsymbol{I}$ is the identity matrix.

For the $\boldsymbol{\xi}_t$, we define a potential function $H_t(\boldsymbol{r})$ that is used to calculate the nullspace motion towards the target,

$$H_t(\boldsymbol{r}) = \frac{1}{2}(\boldsymbol{r} - \boldsymbol{r}_t)^T \boldsymbol{W}_t(\boldsymbol{r} - \boldsymbol{r}_t). \tag{5}$$

Its gradient is written with the overall task Jacobian $\boldsymbol{J}(\boldsymbol{q})$,

$$\frac{\partial H_t(\boldsymbol{r})}{\partial \boldsymbol{q}} = \frac{\partial H_t(\boldsymbol{r})}{\partial \boldsymbol{r}} \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{q}} = \frac{\partial H_t(\boldsymbol{r})}{\partial \boldsymbol{r}} \boldsymbol{J}(\boldsymbol{q}). \tag{6}$$

Then, let α_t be a step width,

$$\boldsymbol{\xi}_t = -\alpha_t (\frac{\partial H_t(\boldsymbol{r})}{\partial \boldsymbol{q}})^T, \tag{7}$$

where $\boldsymbol{W}_t$ is a weighting matrix, $\boldsymbol{r}$ is the current task vector and $\boldsymbol{r}_t$ is the target task vector.

The output of the collision avoidance control is a joint velocity vector.

C. Whole body motion control

For the reaching motion, we use a whole body motion control [17] that uses redundant DOFs with nullspace optimization criteria as,

$$\dot{\boldsymbol{q}}_{wbm} = \boldsymbol{J}^{\#}_{wbm}(\boldsymbol{q})\dot{\boldsymbol{r}}_{task} + \boldsymbol{N}_{wbm}\boldsymbol{\xi}_{wbm}, \tag{8}$$

$$\boldsymbol{N}_{wbm} = \boldsymbol{I} - \boldsymbol{J}^{\#}_{wbm}(\boldsymbol{q})\boldsymbol{J}_{wbm}(\boldsymbol{q}), \tag{9}$$

where $\dot{\boldsymbol{q}}_{wbm}$, $\boldsymbol{J}^{\#}_{wbm}(\boldsymbol{q})$ and $\dot{\boldsymbol{r}}_{task}$ are the joint velocity vector, the pseudo inverse of the task Jacobian and the task velocity vector, respectively.

The matrix $\boldsymbol{N}_{wbm}$ maps the vector $\boldsymbol{\xi}_{wbm}$ into the nullspace that is composed of two criteria,

$$\boldsymbol{\xi}_{wbm} = \boldsymbol{\xi}_{jc} + \boldsymbol{\xi}_{ca}. \tag{10}$$

The $\boldsymbol{\xi}_{jc}$ is a joint limit avoidance cost function,

$$H_{jc}(\boldsymbol{q}) = \frac{1}{2}(\boldsymbol{q} - \tilde{\boldsymbol{q}})^T \boldsymbol{W}_{jc}(\boldsymbol{q} - \tilde{\boldsymbol{q}}), \tag{11}$$

$$\boldsymbol{\xi}_{jc} = -\alpha_{jc} (\frac{\partial H_{jc}(\boldsymbol{q})}{\partial \boldsymbol{q}})^T, \tag{12}$$

where $\boldsymbol{W}_{jc}$ is the weighting matrix, $\tilde{\boldsymbol{q}}$ is the vector of joint centers and α_{jc} is the step width. The $\boldsymbol{\xi}_{jc}$ will keep the joints away from their limits.

The $\boldsymbol{\xi}_{ca}$ is the gradient of a potential function that avoids collisions,

$$\boldsymbol{\xi}_{ca} = -\alpha_{ca}\boldsymbol{J}^{\#}_{ca}(\boldsymbol{q})|_{row,y}(d_{safe} - d), \tag{13}$$

where d_{safe} is a safety distance that is sufficient large so that $d_{safe} - d$ is always positive and α_{ca} is the step width.

D. Integration of whole body motion control and collision avoidance

The computed joint velocity vectors $\dot{\boldsymbol{q}}_{ca}$ and $\dot{\boldsymbol{q}}_{wbm}$ that are outputs of the symmetrical equations Eq. (2) and Eq. (9) are integrated with the function $f(d)$ in Eq. (1).

Function $f(d)$ determines the distribution of the joint velocity vectors,

$$f(d) = \begin{cases} \frac{d - d_a}{d_b - da} & \text{if} \quad d \leq d_a, \\ 0 & \text{otherwise.} \end{cases} \tag{14}$$

If the closest segments are further apart, the whole body motion control $\dot{\boldsymbol{q}}_{wbm}$ as in Eq. (9) has full control of the motion. On the other hand, distance d_b determines a so-called *orange zone* which is always less than d_a. If the closest segments go into this zone, the collision avoidance control $\dot{\boldsymbol{q}}_{ca}$ as in Eq. (2) has full control of the motion. If d is between d_b and d_a, both the collision avoidance control and the whole body motion control affect the robot motion weighted with the function $f(d)$.

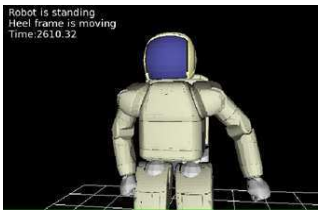
Note that the collision avoidance motion always affects the robot motions even if $d > d_a$ by the nullspace of the whole body motion control. The target reaching motion is also affected by the nullspace of the collision avoidance control even if $d < d_b$.

IV. Experiments

A. Experiments environment

Experiments have been carried out on the humanoid robot ASIMO that has 5 DOFs per arm, 6 DOFs to describe the virtual link between heel and upper body, 3 DOFs for the heel coordinate and 2 DOFs for head movements. Thus, the whole body motion controls 21 DOFs in total. The distance computations for the collision avoidance are done between all segments pairs that can collide as shown in Fig. 1. For instance, the distances from *Right Forearm* and *Right Hand* against *Head*, *Left Shoulder*, *Left Upper Arm*, *Left Forearm*, *Left Hand*, *Chest*, *Body* and *Right Thigh* are computed respectively. The *Left Forearm* and *Left Hand* are computed in the same manner. The distance d is the distance between closest segments. The distance thresholds have been set to $d_a = 4cm$ and $d_b = 1cm$. We use 4 joints, 3 shoulder joints and 1 elbow joint for the collision avoidance of each arm. The joints between the lower arm and the hand are not used since the hand rotation is around the forearm axis only.

If a new target is commanded a linear trajectory between the current and the new target is generated. All computations except vision processing are performed on the robot's embedded computer in real-time. The sampling time for computations of the total control system including the distance computation and the collision avoidance is $5msec$.



(a) Simulation

(b) Robot

Fig. 4. Example of a target that is inside the body.

Fig. 5. Example of arm avoidance. The motion without the collision avoidance on the simulator and with on the robot.

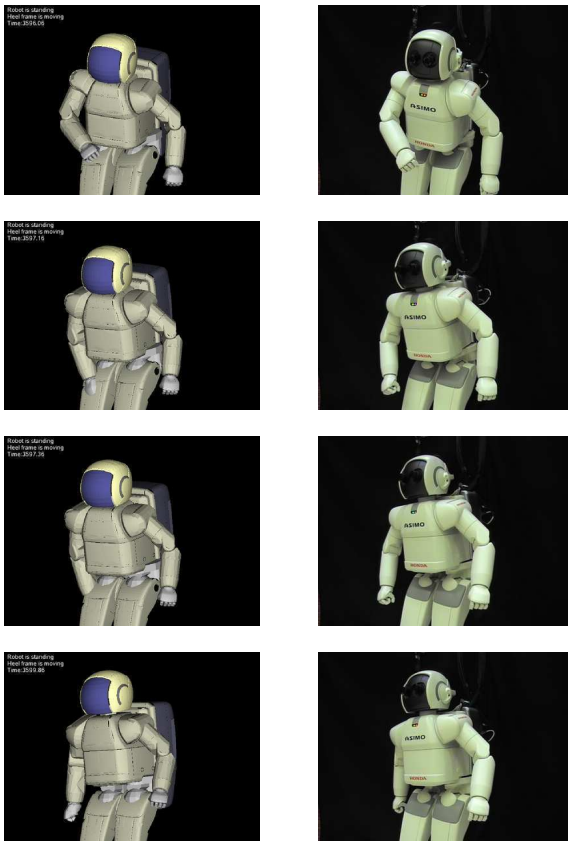


Fig. 6. Example of the body avoidance. The same motion are done both on the simulation (left) and the robot (right) from the top to the bottom. The pictures of the first row show the initial position and of the last row show the target position. The second and the third rows show the motion in between without (left) and with (right) the collision avoidance.

B. Experiments

The following experiments have been conducted. In several experiments, we disabled walking in order to highlight the effectiveness of arm motions.

1) A target is inside the body:
The target of the right arm is inside the body. When the collision avoidance is deactivated, the lower arm collides with the body, as shown in the left part of Fig. 4 (simulation). The right part with the real robot shows that the arm motion stops at the side of the body. The robot moves its body so that the target is reached even if it is inside the body. This is due to the compensation by the whole body motion control.

2) Arm targets lead to arm-arm-collision:
The targets for the arms are static 3D positions in front of the body. They are chosen in such a manner that the arms have to cross and that the arms would collide without collision avoidance. Fig. 5 shows an example. The final posture of the robot is close to the commanded targets but with minimum distance d_b between the forearms.

3) The linear trajectory between the current and the target positions is violating the robot's segments:
In Fig. 6, segments of the robot lie on the trajectory between the current position and the target, but the target is outside of the body. This is a typical case where the robot cannot reach the target without the collision avoidance. The collision avoidance pushes outward by means of the virtual force while the arm limbs would violate the body or the leg. Fig. 7 illustrates the virtual force and the closest distance. The virtual force is generated when the closest distance is less than d_a. At time 3.63sec, the closest distance is switched from *Body* to *Right Thigh*. Finally, the target is reached without colliding.

4) The target is temporarily inside the body while walking: The collision avoidance works while walking as illustrated in Fig. 8. The absolute arm target is in front of the robot when it starts to walk Fig. 8(a). The target for the walking is also in front of the robot but further away. The robot reaches to the target as seen in Fig. 8(b). However, the target for the leg position is still forward, so the robot keeps walking. At some time the arm target is behind of the robot (Fig. 8(c)) and the collision avoidance prevents the arm from penetrating the body. When the walking stops (Fig. 8(d)), the collision avoidance still affects the robot's motion seamlessly.

Note that in this case the leg position moves but it is not a DOF that can be used by the collision avoidance directly or indirectly.

5) Targets are far outside movement range:
The collision avoidance is also safe for extreme postures as shown in Fig. 9. The target for the right arm is far leftward and for the left one is far rightward.

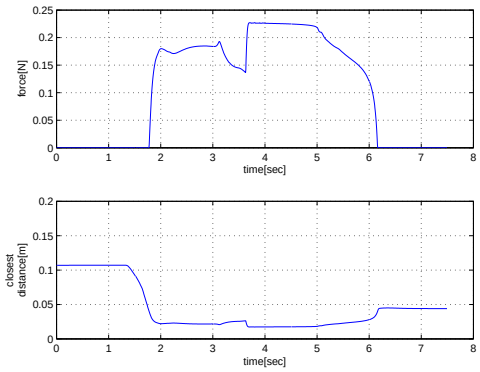
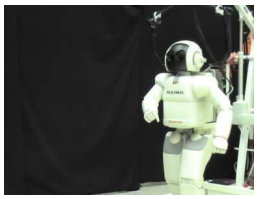
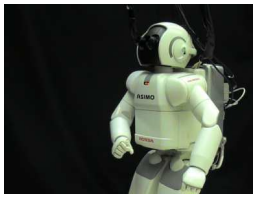


Fig. 7. The virtual force (top) and the closest distance (bottom)

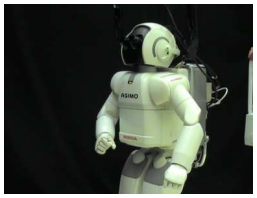


(a) Start walking

(b) Reach a target



(c) Avoid while walking



(d) Stop walking

Fig. 8. Collision avoidance while walking. The target for right arm is 0.8m in front of the body, and the robot is commanded to walk forward 1.5m.

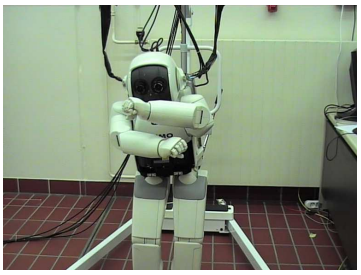
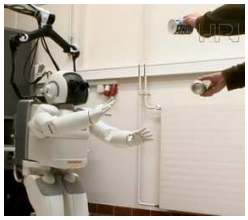
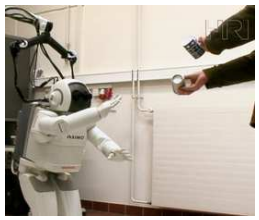


Fig. 9. Extreme posture



(a)



(b)

(c)

(d)

Fig. 10. This is an example of an interactive motion based on vision. (a): The robot tracks two objects (a blue cup and a can) that are in the human's hand. (b): Both targets move counter clockwise in an arc from the robot's point of view and the robot tracks them. (c): Finally, human's arms collide but the robot's don't because of the collision avoidance. (d): Closeup picture of (c).

6) The robot points at objects presented interactively and moving in an unpredictable fashion:
Finally, we tested the collision avoidance in an interactive scenario that is a typical example for unpredictable motions. The scenario is shown in Fig. 10; A human holds objects in front of the robot. The robot continuously points with its hands, their positions are determined by a stereo vision system. The frame rate of the images is about 10Hz and in this experiment, the target positions are also updated with this frequency. The robot has to continuously point and avoid collisions in real-time.

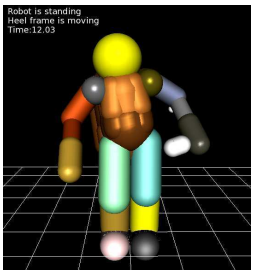
When two targets come into the range of the cameras, the robot reaches for them. This is depicted in Fig. 10(a) to Fig. 10(c). In Fig. 10(c), the robot stops pointing because the closest distance is almost d_b and $f(d) \approx 1$ in Eq. (1).

7) External objects are considered parts of the body (body schema):
External objects can be considered as additional segments that the robot avoids. For instance, in Fig. 11(a), the robot grasps an object that can be considered an additional segment. The robot avoids this segment as if it were one of the robot segments. This corresponds to an extension of the robot's *body schema* [18].

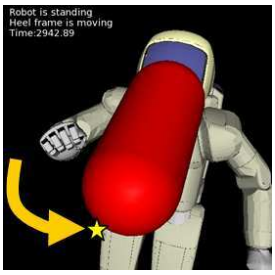
8) A virtual object is avoided to realize an occlusion avoidance:
It is also possible to define *virtual objects* in space as a robot segment. We propose to use this for applications like "occlusion avoidance" as shown in Fig. 11(b). One of the major problems when robots grasp an object are occlusions (the hands hide the target objects). We defined a virtual segment between the robot head and the target so that hands do not enter the center field of view. Just before the robot grasps the object, the virtual segment is switched off. By this method, trajectories do not hinder visual tracking of targets.

V. Discussion

The collision avoidance works in different situations, not only while standing but also while walking. If the target cannot be reached, the arm moves to the position that is closest to the target. The collision avoidance motion is compensated by the target reaching motion, which is composed of 2 parts: the task space motion of the whole body motion control and the nullspace motion of the collision avoidance control. In particular, the collision avoidance uses only 1 DOF for each



(a) body schema



(b) Occlusion avoidance

Fig. 11. (a): The robot avoids the object that is attached to the hand. (b): The virtual object (red SSL) is generated between the head and the target (yellow star) so that the arms movements don't violate the gaze line.

arm and the other DOFs can be used for target reaching motions in nullspace.

The continuous task priority changing is realized by the distance between closest points. The collision avoidance works not only for robot segments but also for external objects, that can be dynamically modified or switched on and off. It is possible to avoid any objects by superposing the robot's motions.

The collision avoidance may have 2 possible problems. One is local minima since it reacts to obstacles that enter the yellow zone. This should be solved on a planning level that takes optimal trajectories into account. In other words, the planning method should handle global criteria in stable environments while the real-time collision avoidance assures safety and handles the local criteria in highly dynamic environments by superposing the planned trajectory.

The other problem is discontinuities in motions because the collision model and the structure of the robot are not convex and the closest segments pair is sometimes switched. This can be solved by computing them based on not only one segment pair but also more segment pairs, for instance, to compute all segment pairs that stay in the yellow zone.

VI. Conclusion and Outlook

We realized a real-time collision avoidance system on a humanoid robot running on on-board embedded computers. It works in a highly dynamic situations such as crossing arms, walking to arbitrary targets and interactive motions. The robot moves to a given target while avoiding collisions without stopping. The priority between the target reaching motion and the avoidance motion is changed depending on the distance between closest segments. The method can be extended to objects that the robot grasps (body schema) and it can be applied to occlusion problems.

We are currently investigating closest segments computations and the trajectory planning method that integrates with the real-time collision avoidance in order to generate non-discontinuous and globally optimal trajectories in uncertain environments.

References

[1] T. Lozano-Pérez and M. A. Wesly, "An algorithm for planning collision-free paths among polyhedral obstacles," in *Communications of the ACM*, vol. 22, no. 10, 1979, pp. 560–570.

[2] O. Khatib, "Real-time obstacle avoidance for manipulations and mobile robots," in *The international Journal of Robotics Research*, 1986.

[3] J. Kuffner, K. Nishiwaki, S. Kagami, Y. Kuniyoshi, Masayuki, Inaba, and H. Inoue, "Self-collision detection and prevention for humanoid robots," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2002.

[4] J. Kuffner, S. Kagami, K. Nishiwaki, M. Inaba, and H. Inoue, "Dynamically-stable motion planning for humanoid robots," in *Autonomous Robots*, vol. 12, no. 12, 2002, pp. 105–118.

[5] Y. Nakamura, *Advanced Robotics: Redundancy and Optimization*. Addison Wesley Publishing Company, 1991.

[6] H. Hanafusa, T. Yoshikawa, and Y. Nakamura, "Redundancy analysis of articulated robot arms and its utilization for task with priority," in *SICE*, vol. 19, no. 5, 1983, pp. 421–426.

[7] A. A. Maciejewski and C. A. Klein, "Obstacle avoidance for kinematically redundant manipulators in dynamically varying environments," in *The international Journal of Robotics Research*, vol. 4, no. 3, 1985, pp. 109–117.

[8] O. Brock, O. Khatib, and S. Viji, "Task consistent obstacle avoidance and motion behavior for mobile manipulation," in *Proceedings of the IEEE International Conference of Robotics and Automation*, 2002.

[9] F. Seto, K. Kosuge, and Y. Hirata, "Self-collision avoidance motion control for human robot cooperation system using robe," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005.

[10] F. Seto, Y. Hirata, and K. Kosuge, "Motion generation for human-robot cooperation considering range of joint movement," in *Proceedings of the IEEE-RAS International Conference on Intelligent Robots and Systems*, 2006.

[11] H. Sugiura, M. Gienger, H. Janssen, and C. Goerick, "Real-time self collision avoidance for humanoids by means of nullspace criteria and task intervals," in *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*, 2006.

[12] C. Ericson, *Real-time collision detection*. The Morgan Kaufman Publishers, 2005.

[13] G. van den Bergen, *Collision Detection in Interactive 3D Environments*. The Morgan Kaufman Publishers, 2004.

[14] M. Deloura, *Game programming Gems2*. Charles River Media, Inc., 2001.

[15] E. Lengyel, *Mathematics for 3D Game programming & Computer graphics second edition*. Charles River Media, Inc., 2004.

[16] E. Larsen, S. Gottschalk, M. C. Lin, and D. Manocha, "Fast proximity queries with swept sphere volumes," in *Technical report TR99-018, Department of Computer Science, University of N. Carolina, Chapel Hill.*, 1999.

[17] M. Gienger, H. Janssen, and C. Goerick, "Task-oriented whole body motion for humanoid robots," in *Proceedings of the IEEE-RAS International Conference on Humanoid Robots*, 2005.

[18] J. Paillard, "Body schema and body image - a double dissociation in deafferented patients," in *In Motor Control, Today and Tomorrow, eds. G. N. Gantchev, S. Mori and J.Massion. Bulgarian Academy of Science Sofia: Academic Publishing House.*, 1999.